

Iljo De Poorter

Iljo De Poorter. OOSD

basic info/inleiding

Abstractie, In werkelijkheid is dit essentieel. Niveau van detail is nodig.

•Objecten

Objecten zijn dingen die een bepaalde betekenis hebben. Deze kunnen we van elkaar onderscheiden, ze hebben ook eigenschappen en kunnen gebruikt worden. Zolang dat de code gecompileerd is maar niet aan het "runnen" worden de objecten niet aangemaakt; dat is enkel in runtime.

•Klassen

De verschillende objecten met gelijkaardige eigenschappen worden gegroepeerd in een klasse. In databases is dit een entiteitstype. De beschrijvingen van deze dingen zijn belangrijk, Bv. Wat kan ik doen met een bordstift.

Horloges— Een lijst horloges wordt geabstrakteerd naar gewoon polshorloges. Die dan wel eigenschappen hebben zoals kleur, materiaal. 1 object is de instantie van de klasse. 1 ding van de klasse wordt gedefinieerd. Synoniem voor object. Klasse Polshorloge(algemeene omschrijving), Die allemaal gemeenschappelijke eigenschappen hebben.

klassen en objecten

Klassen zijn de blauwdrukken van objecten. Objecten zijn de instanties van klassen. Alles IN OOP IS EEN OBJECT. Een scanner is Bv een object van de klasse Scanner.

Identiteit van een object is belangrijk, 2 objecten zijn nooit hetzelfde. Ze kunnen wel dezelfde eigenschappen hebben. Alles is een object, Een object wordt beschreven door een klasse en een object is een instantie van een klasse

•toestand

Objecten zijn instanties van klassen.

Een toestand van een object zijn alle eigenschappen, maar ook de waarde. Zo kan een eigenschap kleur zijn, maar de waarde van die kleur is dan bv. rood. Daarom is de toestand van belang.

Identiteit = unieke naam van een object. Met hulp van de constructoren.

•DB vs OOP

Entiteitstype = klasse Entiteit = object/instantie

Klassen hebben een eigenschap en een gedrag. (De omschrijving van wat elk object van de klasse gemeenschappelijk heeft).

•Attribuut

Attribuut = 1 eigenschap en een klasse. Een attribuut zal een bepaalde waarden hebben, Dit doen we door 2 dingen bij te houden, de NAAM van het attribuut en het TYPE van deze. Let wel op met de naamgeving. De naam moet altijd een werkwoord in een gebiedende wijze zijn

Het type is het soort waarde dat het attribuut kan aannemen. Bv. String, int, double, float, boolean, char, byte, short, long.

Visibiliteit, geeft aan of het attribuut zichtbaar is voor andere objecten. Dit is belangrijker voor de encapsulatie. Dit is een van de 4 principes van OOP.

•Voorbeeld

Types van objecten is zeer belangrijk, vanaf dat we het hebben over een object moeten we meteen nadenken welk type dit object zal worden. Zoals Bv. Float, dubbele, int. //table Kauwgom Kleur String Hoogte double Prijs double Aantalballen int //endtable

•In de praktijk

Speciale typen methoden CONSTRUCTORS-; Om een instantie te maken van een bepaalde klassen

Een constructor heeft een signatuur, deze heeft altijd een naam, DIE EXACT HETZELFDE GESCHREVEN IS ALS DE KLASSE. Een constructor heeft parameters met critieke waarden.

GEEN RETURNtype.

Een constructor is een methode die een object maakt van een bepaalde klasse.

new Scanner(System.in); -; Dit is een constructor van de klasse Scanner. En System.in is de parameter.

•Klassen en Objecten

Objecten hebben een gedrag – diensten die ze kunnen aanbieden

–Digen die het object kan doen

Objecten Sturen boodschappen naar objecten –Objecten vragen aan andere objecten om iets te doen

–Zo doen objecten beroep op elkaar

Een gedrag geeft 1 methode, 1 methode heeft maar 1 taak(UNIX GANG) Methodes mogen geen code bevatten die niet tot hun taak behoort.

Een constructor is geen methode, want een Methode heeft 3 delen, een Constructor maar 2.

De signatuur van een methode heeft ook een naam, heeft ook parameters en heeft ook een returntype. maar wel altijd een return type.

Het stuk Logica van een methode,

Stelt objecten in staat om op een correcte wijze te reageren op boodschappen

Voorlopig gaan we niet verder inzoomen op de logica.

•Voorbeelden signatuur

Naam, Bv. Veranderstatus

Parameters, Bv. bij verandergraote is heeft een % graote dus het is een float Parameters bij een auto die versneld. Het is een int extrasnelheid genaamd versnelling, en het is een int.

Returntype, Bv. bij verandergraote is het een void, want het verandert de graote van het object.

Omschrijving van de reactie, wat moet er gebeuren in de logica zelve.

•Get and Set

Getters en setters zijn methodes die de toestand van een object kunnen veranderen.

Getters = geef waarde van 1 attribuut Naam, getAttribuut Altijd return; Nooit de waarde aanpassen =; geen parameters

Setters = verander waarde van 1 attribuut Naam, setAttribuut niets retourneren

uml

Unified Modeling Language Deze word gebruikt om de structuur van een programma te beschrijven.

We maken maar 1 diagram voor OOSD. Elke klasse noemen we met een Hoofdletter, elk object is singelvoudig, dus steen ipv stenen.

•get and set In UML

Public getter. (Hiervan zijn er enorm veel, door deze uit te tekenen in UML vervuult dit het diagram.) ;

Public setter ; Als we een public setter toevoegen moeten we het er altijd opschrijven, want dit is niet standaard.

Packages in JAVA vanaf NU—

- CUI, alles met interactie met de gebruiker. Scanner, Sysout, enz..
- domein. Alle domeinklassen, Hier komt alle logica. Hier mag GEEN main method. Geen scanner, geen Sysout.
- testen. Hier plaatsen we de test packages.